Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

Bescheinigung

Certificate

Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. **Patent application No.** **Demande de brevet n°**

02368138.0

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk

DEN HAAG, DEN
THE HAGUE, 28/03/03
LA HAYE, LE

Europäisches Patentamt | European Patent Office | Office européen des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.: 02368138.0
Demande n°:

Anmeldetag:
Date of filing: 19/12/02
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):
INTERNATIONAL BUSINESS MACHINES CORPORATION
Armonk, NY 10504
UNITED STATES OF AMERICA

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
A packet unstopper system for a parallel packet switch

In Anspruch genommene Prioriät(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat: State: Pays:	Tag: Date: Date:	Aktenzeichen: File no. Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:
/

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BG/BE/CH/CY/CZ/DE/DK/EE/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

A PACKET UNSTOPPER SYSTEM FOR A PARALLEL PACKET SWITCH

Field of the Invention

The present invention relates to high speed switching of data packets in general, and particularly relates to a system and method to control the saturation of egress buffers that store data packets switched through a plurality of independent switching planes of a parallel packet switch architecture

Background of the Invention

DWDM, which stands for Dense Wavelength Division Multiplexing, by merging onto a single optical fiber many wavelengths, is making available long-haul fiber-optic data communications links of huge aggregate capacity. Each wavelength is an independent communications channel which typically operates at OC48c i.e.: 2.5 Giga or 10^9 bits per Second (Gbps), OC192c (10 Gbps) and in some systems at OC768c (40 Gbps). These rates are part of a family of rates and formats available for use in optical interfaces, generally referred to as SONET, which is a standard defined by the American National Standards Institute (ANSI) of which there exists an European counterpart, mostly compatible, known as SDH (Synchronous Digital Hierarchy). Thus, at each node of a network, the data packets or cells carried on each DWDM channel must be switched, or routed, by packet-switches that process and then switch packets between different channels so as to forward them towards their final

destination. If, ideally, it would be desirable to keep the processing of packets in the optical domain, without conversion to electronic form, this is still not really feasible today mainly because all packet-switches need buffering that is not yet available in an optical form. So packet-switches will continue to use electronic switching technology and buffer memories for some time to come.

However, because of the data rates as quoted above for individual DWDM channels (up to 40 Gbps) and the possibility of merging tenths, if not hundredths, of such channels onto a single fiber the throughput to handle at each network node can become enormous i.e., in a multi Tera or 10^{12} bits per second range (Tbps) making buffering and switching, in the electronic domain, an extremely challenging task. If constant significant progress has been sustained, for decades, in the integration of always more logic gates and memory bits on a single ASIC (Application Specific Integrated Circuit), allowing to implement the complex functions required to handle the data packets flowing into a node according to QoS (Quality of Service) rules unfortunately, the progress in speed and performance of the logic devices over time is comparatively slow, and now gated by the power one can afford to dissipate in a module to achieve it. Especially, the time to perform a random access into an affordable memory e.g., an imbedded RAM (Random Access Memory) in a standard CMOS (Complementary MOS) ASIC, is decreasing only slowly with time while switch ports need to interface channels having their speed quadrupling at each new generation i.e., from OC48c to OC192c and to OC768c respectively from 2.5 to 10 and 40 Gbps. For example, if a memory is 512-bit wide allowing to store or fetch, in a single write or read operation, a typical fixed-size 64-byte (8-bit byte) packet of the kind handled by a switch, this must be achieved in less than 10 Nano or 10^{-9} second (Ns)

for a 40 Gbps channel and in practice in a few Ns only in order to take care of the necessary speed overhead needed to sustain the specified nominal channel performance while at least one store and one fetch i.e., two operations, are always necessary per packet movement. This represents, nowadays, the upper limit at which memories and CMOS technology can be cycled making the design of multi Tbps-class switch extremely difficult with a cost-performance state-of-the-art technology such as CMOS, since it can only be operated at a speed comparable to the data rate of the channel they have to process.

Hence, to design and implement a high capacity packet-switch (i.e.: having a multi Tbps aggregate throughput) from/to OC768c (40 Gps) ports a practical architecture, often considered to overcome the above mentioned technology limitation, is a parallel packet switch (PPS) architecture. It is comprised of multiple identical lower-speed packet-switches (100) operating independently and in parallel, as sketched in figure 1. In each ingress port adapter, such as (110), an incoming flow of packets (120) is spread (130), packet-by-packet, by a load balancer across the slower packet-switches, then recombined by a multiplexor (140) in the egress part of each port adapter e.g., (150). As seen by an arriving packet, a PPS is a single-stage packet-switch that needs to have only a fraction of the performance necessary to sustain the port (125) data rate. If four planes (100) are used, as shown in figure 1, they need only to have one fourth of the performance that would otherwise be required to handle a full port data rate. More specifically, four independent switches, designed with OC192c ports, can be associated to offer OC768c port speed, provided that ingress and egress port adapters (110, 150) are able to load balance and recombine the packets. This approach is well known from the art and sometimes referred to as 'Inverse

Multiplexing' or 'Load Balancing'. Among many publications on the subject one may e.g., refer to a paper published in Proc. ICC'92, 311.1.1-311.1.5, 1992, by T. ARAMAKI et al., untitled '*Parallel "ATOM" Switch Architecture for High-Speed ATM Networks*' which discusses the kind of architecture considered here.

The above scheme is also attractive because of its inherent capability to support redundancy. By placing more planes than what is strictly necessary it is possible to hot replace a defective plane without having to stop traffic. When a plane is detected as being or becoming defective ingress adapter load balancers can be instructed to skip the defective plane. When all the traffic from the defective plane has been drained out it can be removed and replaced by a new one and load balancers set back to their previous mode of operation.

Thus, if PPS is really attractive to support multi-Gbps channel speeds and more particularly OC768c switch ports it remains that this approach introduces the problem of packet re-sequencing in the egress adapter. Packets from an input port adapter (110) may possibly arrive out of sequence in a target egress adapter (150) because the various switching paths, here comprised of four planes (100), do not have the same transfer delay since they run independently thus, can have different buffering delays. A discussion and proposed solutions to this problem can be found, for example, in a paper by Y.C. JUNG et al., 'Analysis of out-of-sequence problem and preventive schemes in parallel switch architecture for high-speed ATM network', published in IEEE Proc.-Commun., Vol. 141, No. 1, February 1994. However, this paper does not consider the practical case where the switching planes have also to handle packets on a priority basis so as to support a Class of Service (CoS) mode of operation, a mandatory feature in all recent switches which are assumed

to be capable of handling simultaneously all sorts of traffic at nodes of a single ubiquitous network handling carrier-class voice traffic as well as video distribution or just straight data file transfer. Hence, packets are processed differently by the switching planes depending on the priority tags they carry, and may incur very different transit delays depending on which switching plane they have been sent. As each ingress adapter makes its own decision on how it load balance the traffic among the different switching planes depending on the flow control information it receives, it may happen that not all switching planes are loaded in the same way, thus creating different delays for packets transmission over different switching planes. This does no longer comply with the simple FCFS (First-Come-First-Served) rule assumed by the above referenced paper and forces egress adapters to readout packets as soon as they are ready to be delivered by the switching planes after which they can be resequenced on a per priority basis taking in account the fact that packets coming from same source with same priority may have very different transit time when crossing the different switching planes.

Different mechanisms have been proposed to perform the resequencing of packets within a Parallel Packet Switch. However, all of them must face the difficulty that, due to the fact that switching planes may not be instantly identically loaded, in particular because of the multiple priorities in use, two packets sent in sequence by the same source on two different switching planes may incur very different transit delay until they reach the same egress adapter. Especially, low priority packets can easily be trapped in individual switching planes because higher priority packets takes precedence. This clearly may create situations where a packet sent as second by a source, is received first in an egress adapter where it has to be kept in buffer, until

first packet is finally received. Only then, a request can be posted to the egress scheduler which must authorize sucessively both packets to leave the egress buffer on external interface.

In egress buffer, possibly many incomplete flows waiting for trapped packets may thus accumulate taking up space. Depending on the size of the buffer used to store packets in egress adapter, this may lead rapidly to an unacceptable congestion situation that would require to discard those of the packets already switched while missing ones are trapped in undetermined switching planes. Also, this may severely impact the end to end jitter, from ingress to egress line interface.

Object of the Invention

Thus, it is a broad object of the invention to overcome the difficulties mentioned here above in order to make feasible a PPS architecture in which variable delays can be experienced in the individual switching planes while supporting priority classes of unicast and multicast traffic in view of the implementation of a multi-Tbps switch.

It is another object of the invention to avoid egress buffer saturation and bound transit delays by providing an efficient and novel mechanism that allow to identify unambiguously which switching planes are responsible of the buffer saturation and to unstop the flow of data packets for them.

In a preferred embodiment, the invention applies in a parallel packet switch architecture having at least one egress adapter arranged to temporarily store within an egress buffer data packets switched through a plurality of independent switching planes. Each data packet belongs to a data packet flow wherein the data packets are sequentially numbered with a packet sequence number. The invention claims a system for controlling the egress buffer saturation and which comprises for each data packet flow:

means for comparing the number of data packets 'WPC' temporarily stored within the egress buffer to a predefined threshold value 'WPCth';

means for storing the packet sequence number 'PSNr' of a last received in-sequence data packet, and each highest packet sequence number 'HPSNj' respectively received through the plurality of switching planes; and

means coupled to the comparing means and to the storing means for determining at least one switching plane among the plurality of switching planes on which to unstop the flow of data packets by comparing the last received in-sequence packet sequence number 'PSNr' to each highest packet sequence number 'HPSNj' when the number of data packets 'WPC' exceeds the predefined threshold value 'WPCth'.

Further objects, features and advantages of the present invention will become apparent to the ones skilled in the art upon examination of the following description in reference to the accompanying drawings. It is intended that any additional advantages be incorporated herein.

Brief Description of the Drawings

Figure 1 shows a conceptual view of a parallel packet switch system to implement the invention;

Figure 2 is a block diagram showing the main components of a preferred embodiment of the invention;

Figure 3 illustrates an incomplete numbered sequence of data packets;

Figure 4 details the unstop block diagram shown on figure 2;

Figure 5 is a flow chart of the incoming packet process in the egress adapter;

Figure 6 is a flow chart of the outgoing packet process in the egress adapter;

Figure 7 is a flow chart of the process to determine the switching plane to be stopped;

Figure 8 is a schematic view to illustrate the wrapping of the source counters.

Detailed Description of the Preferred Embodiment

Figure 2 shows a functional view of a preferred PPS architecture to operate the invention. For sake of clarity, only one ingress adapter (200) is shown interfacing a plurality of switching planes (planes A to X under block 250) over which an incoming traffic arriving on ingress interface (290) is load balanced by a load balancer circuit (205). The skilled man will easily understand throught the reading of the entire description that all functional principles are to be generalized to a plurality of ingess adapters.

Although the invention does not assume anything on which method is used to perform the re-sequencing of data packets in the egress adapter, however, all packets have to carry an identification of the source which has emitted them (i.e. an ingress adapter identifier among the many ones within the switch fabric), together with a PSN (Packet Sequence Number). It is to be appreciated that there are as many PSN generators as there are possible destinations and priorities in one ingress adapter. PSN is e.g., incremented by 1 with each new packet destined for an output port at a given priority. One possible implementation is shown in block (210), which takes benefit of a Virtual Output Queuing (VOQ) organization (230), a common feature of modern packet switches, which allows to avoid destination head of line blocking. VOQ is not part of the present invention and is not further described since it is a well-known technique by those skilled in the art. For each VOQ, there is an associ-ated counting device (2100 to 2163) which generates a PSN. Counting devices may operate independently of each other, each one corresponding to a unique packet flow identified by its source, its destination, and its priority. Each VOQ feeds a packet scheduling function (220) in each ingress

port-adapter, which selects the waiting incoming packets to be switched. On egress adapter, packets received through the various planes (250), are temporarily stored in an egress buffer (265). An unstop logic block (275) is coupled to the egress buffer and to a packet scheduler (280) to determine if a switching plane is to be unstopped. The unstop logic is arranged to have access to:

- The PSN of the incoming packets from all switching planes, per source, per priority; and
- The PSN of the 'last ready packet for scheduling' (PSNr) by source and priority.

The 'last ready packet for scheduling' relates to the last packet for which a continuous sequence has been recovered by the packet resequencing function (270)and is thus available for sheduling (i.e. Ready for leaving the egress adapter). This will further detailed with reference to figure 3.

Whichever resequencing mechanism is used, each egress adapter is equipped with an output scheduler (280), role of which is to select, at each packet cycle, what is the next packet, temporarily stored in the egress buffer (265), due to leave the egress adapter. The ingress and egress packet scheduling are mechanisms beyond the scope of the invention and are not further discussed other than to mention that their role is normally to serve the waiting packets of highest priorities first while, for each priority, maintaining fairness between the sources of traffic. These are standard functions in switch port-adapters. Packet scheduling (220, 280) and VOQ's (230) are not part of the invention which does not require any specific behaving from these elements to operate as specified in the rest of the description. Figure 2 and following figures illustrate the invention on a preferred implementation assuming that the switch is a 64-port switch, thus with VOQ's having 64 destinations (0-63) per priority.

Generally, the switch port-adapters have a standard line or NP (network processor) IN and OUT interface (290) e.g., such as the ones defined by the Network Processing Forum (NPF), 39355 California Street, Suite 307, Fremont, CA 94538.

Figure 3 shows an example of PSN's belonging to a same flow of packets i.e., packets from a same source, towards a same destination at a same priority, which have been switched over any switching planes A to X (250). Packets stamped with PSN's referenced N, N+1, N+2 (grouped under block 310) are in sequence and may be scheduled at any time to leave on egress interface (290). Packets stamped with PSN's referenced N+4, N+5, N+6 (grouped under block 320) are also in sequence but cannot be scheduled because the packet referenced N+3 has not been received yet. Similarly, packets stamped with PSN's from N+8 to N+11 (grouped under block 330) are in sequence but cannot be scheduled as long as packets N+3 and N+7 have not been received. A waiting packet counter (WPC) 340, counts all data packets stored in the egress buffer that belong to a same flow. This count is done irrespectively of the fact that packets are in sequence and thus can be scheduled over the egress interface line, or are blocked because there is one or more missing packets. In the present example, packet stamped with PSN equal to N+2 is the 'last received in sequence' packet.

As shown in figure 3, for the sake of simplicity, the invention is exemplified assuming that packets are numbered at source with an increasing complete sequence of numbers N, N+1, N+2, etc.. However, the invention does not preclude the use of other methods like a decreasing sequence of numbers or any other ordered incomplete sequences of numbers. The only asssumption that must hold is that egress adapter must be capable of determining, whichever numbering or ranking method is adopted, from the received PSN sequences, when

packets are missing (as this is indeed the case above for missing packets n+3 and n+7). Also, it is to be appreciated that terms like 'highest' is to be interpreted in light of the numbering sequence used to illustrate the invention i.e., an ascending sequence. In this context 'highest' means also the latest numbered packet received in the egress adapter from one switching plane.

Figure 4 shows the resources required, in a preferred embodiment of the invention, by the unstop logic block (275) of figure 2. For each flow of data packets, characterized in an egress adapter by its source and priority, there is one set of memory devices e.g., registers (410), used to store:

- The WPC as already discussed above (340) and which indicates the total number of packets stored in the egress buffer for this flow. Packets can be in sequence or not.
- The PSNr, which holds the sequence number of the 'last received in sequence' packet as discussed in figure 3.
- A set of 'Highest Packet Sequence Number' HPSN, one per switching plane. Six HPSN's are examplified (HPSN1 to HPSN6) assuming that the Parallel Packet Switch structure is made, in this example, of six switching planes. Each HPSN's register remembers, per data flow, what is the highest packet sequence number received through the corresponding plane.

Figure 5 describes the algorithm performed within the unstop logic function when a new packet is received by an egress adapter from any switching plane. As previously, to illustrate the invention the following description assumes there are six switching planes. It must be clear however that the invention can be practiced with any other number of switching planes.

Each new packet received from any switching plane j belongs to one flow, identified by its source Sn and priority Pn. Moreover, each incoming packet carries a PSN referred to as PSNi (box 500). The flow identification (by the source Sn and the priority Pn identifiers) allows to retrieve (step 510) from the associated register (410) shown in figure 4 the current status of the flow as characterized by the number of waiting packets in egress buffer for this flow (WPC), the value of the 'last received in sequence' PSNc, and the values of the highest packet sequence numbers received over the six switching planes HPSN1 to HPSN6. Then, HPSNj value of switching plane j on which packet has just arrived is updated (step 520) with the packet sequence number of the received data packet. This rests on the assumption that switching planes never introduce (by design) any disordering in the delivery of packets thus, deliver packets from a same source at a same priority in the exact same order as received from an ingress adapter so that HPSNj is always higher than a previous received one. On next step 530, WPC is incremented. This ends the incoming packet process (540).

Figure 6 describes the operations performed when a packet leaves the egress adapter over the NPF interface (290). At each outgoing packet (box 600), the WPC of the flow to which the outgoing packet belongs and which is identified by Sn and Pn, is retrieved (step 610). Then, WPC is decremented by one (620) indicating that there is one packet less in the egress buffer for that flow. This ends the outgoing packet process (630).

Figure 7 describes the process to determine which switching plane(s) should be unstopped. For each flow, the current WPC is periodically compared to a threshold value

WPCth (box 700). This threshold value is a fixed value, which depends on the size of the egress buffer, the number of switching planes, the number of priorities, and depends in general of the physical characteristics of a particular implementation. Then, when current waiting packet count is lower than WPCth (branch 702), no action is performed until next comparison. When the current waiting packet count WPC equals or is higher than WPCth (branch 701), then each HPSN value for switching planes 1 to 6 (HPSN1 to HPSN6), related to this flow, is retrieved from the corresponding memory device (410) and is compared to the PSN of 'last received in sequence' packet i.e., PSNr (boxes 710, 720, 730, 740, 750 and 760 respectively for switching planes 1 to 6).

Next, if for all switching planes, PSNr is equal to or lower than HPSNj i.e., HPSN1 to HPSN6 (branches 712, 722, 732, 742, 752 and 762 respectively), it means that on all switching planes, data packets have arrived with a packet sequence number higher than the last packet ready for scheduling. However, because PSNr has a lower value it can only mean that resequencing is stuck with PSNr, waiting for the next in sequence i.e., PSNr + 1 (since numbering is assumed to be a complete ascending sequence in the example used to illustate the invention), which packet has never arrived, while packets with PSN value higher than PSNr+1 have arrived on all switching planes (box 770). Although switching planes are normally loss less, it may seldom happen that packets are lost between ingress adapter and egress adapter. Reasons may be transmission error, possibly leading to a corrupted packet. Depending on the level of protection within the system, which is beyond the scope of the present invention, the corrupted packet may be discarded or misrouted, this action leading to the above mentioned blocking in the resequencing mechanism. In such a case, there is nothing else to do than to unblock the resequencing algorithm by

incrementing to the next number in sequence (e.g., by +1) the current value of PSNr (box 775). No unstop action is performed and there is no further processing (box 790) other than, as an option, reporting to the device in charge that a packet was lost.

If PSNr is higher than HPSNj in all the switching planes, i.e., HPSN1 to HPSN6 (branches 711, 721, 731, 741, 751 and 761 respectively), all planes are selected (boxes 713, 723, 733, 743, 753 and 763). This means that, on none of the switching planes a packet has arrived with a PSN higher than the last packet ready for scheduling. PSNr has the highest value. Hence, it can only mean that resequencing has gone until last received value PSNr. Then, all packets as counted in WPC, can be scheduled. For reasons that are beyond the scope of present invention, these waiting packets have not been forwarded on egress interface and have accumulated above WPCth. What to do in this case is highly dependent on the design and system implementation choices of the switching node where the invention is used. No unstop action (box 780) needs to be performed and there is no further process (790) other than, possibly, reporting to the device in charge that packets, yet in sequence, are accumulating.

If for at least one switching plane, but not all of them, PSNr is higher than HPSNj i.e., HPSN1 to HPSN6 (as previously, branches 711, 721, 731, 741, 751 and 761 respectively), then corresponding switching planes are selected (713 to 763 respectively). For those of the switching planes that are selected (box 713,723,733,743,753 and 763) means that no packet has arrived with a packet sequence number higher than the last packet ready for scheduling, while on the non selected ones (there is at least one), packet(s) have arrived with a higher sequence number higher. Then, it means that resequencing is stuck because packets are

expected on those selected switching planes. Typically, this may be caused by low priority packets being blocked by higher priority packets inside selected switching planes. In which case, unstop process is to be triggered on the selected switching planes (box 785), which will eventually unblock the resequencing mechanism. There is no further process (box 790).

The unstop process will not be further described as it is application specific and may be implemented by those skilled in the art in many different ways depending on system characteristics. The mechanism described above allows to determine, without ambiguity, which switching plane(s) have caused the resequencing issue for the related flow. The unstop process provides a means to trigger, in switching planes, the retrieving of packets. The retrieving process itself is highly dependent on the design of the switching planes and is beyond the scope of the invention. This process is implementation dependent. It may or not have an impact on delay and jitter of other packets not belonging to the related flow and being switched over the identified switching plane.

Figure 8 briefly discusses the problem of the wrapping (800) of the counters used to rank packets at ingress or at egress. Those counters have a finite length thus, whichever their counting capacity the problem of their wrapping must be solved. The invention assumes that those counters have one bit more (810) than what is necessary to number the packets. For a given application the counting capacity (820) must be determined so that the oldest numbered packet still waiting in the egress buffer (830) cannot be wrongly compared with a new arriving packet (of the same source with the same priority) because the counter used in the source has wrapped in the mean time. Once this value has been

determined the invention assumes that the counters are all made one bit wider so that numbering of waiting packets cannot span on more than one counter wrapping boundary (850). Then, it is easy to take care of the counter wrapping. One solution consists in detecting the first occurrence of a readout packet number for which MSB (most significant bit) is found to be 0 (860) after a series of ones, in which case comparison of MSB's must be toggled.

Claims

1. In a parallel packet switch architecture having at least one egress adapter (260) arranged to temporarily store within an egress buffer data packets switched through a plurality of independent switching planes (250), each data packet belonging to a data packet flow wherein the data packets being sequentially numbered with a packet sequence number (PSN), a system for controlling the egress buffer saturation comprising for each data packet flow:

   means (340) for comparing the number of data packets 'WPC' temporarily stored within the egress buffer to a predefined threshold value 'WPCth';

   means (410) for storing the packet sequence number 'PSNr' of a last received in-sequence data packet, and each highest packet sequence number 'HPSNj' respectively received through the plurality of switching planes; and

   means (275) coupled to the comparing means and to the storing means for determining at least one switching plane among the plurality of switching planes on which to unstop the flow of data packets by comparing the last-received in-sequence packet sequence number 'PSNr' to each highest packet sequence number 'HPSNj' when the number of data packets 'WPC' exceeds the predefined threshold value 'WPCth'.

2. The system of claim 1 further comprising at least one ingress adapter (200) coupled to the plurality of switching planes, the at least one ingress adapter comprising means (210) to sequentially number the data packets within

each flow and means to identify each data packet within each flow by an ingress adapter identifier (Sn) and by a priority level identifier (Pn).

3. The system of anyone of claims 2 wherein the determining means further comprises means for pointing to the storing means using the ingress adapter identifier (Sn) and the priority level identifier (Pn) of each data packet flow.

4. The system of claims 2 or 3 wherein the at least one ingress adapter further comprises means (205) for load balancing the data packets within each flow among the plurality of switching planes.

5. The system of anyone of claims 1 to 4 wherein the at least one egress adapter further comprises means (270) for resequencing the data packets temporarily stored within the egress buffer for each flow of data packets.

6. The system of claim 5 wherein the at least one egress adapter further comprises means (280) for outputting the resequenced data packets from the egress buffer.

7. The system of anyone of claims 1 to 6 wherein the comparing means further comprises means for counting the number of data packets 'WPC' temporarily stored within the egress buffer for each flow of data packets.

8. The system of claim 7 wherein the counting means further comprises means for decrementing the number of data packets 'WPC' temporarily stored within the egress buffer when a resequenced data packet is output from the egress buffer.

9. The system of anyone of claims 1 to 8 wherein the data packets are numbered with an increasing sequence of data packets numbers.

10. A computer-like readable medium comprising instructions for carrying out a method to operate the system of anyone of claims 1 to 9.

A PACKET UNSTOPPER SYSTEM

FOR A PARALLEL PACKET SWITCH

Abstract

A system for controlling the egress buffer saturation of an egress adapter in a parallel packet switch architecture is disclosed. The egress adapter is arranged to temporarily store within an egress buffer data packets that have been switched through a plurality of independent switching planes. Each data packet belongs to a data packet flow wherein the data packets are sequentially numbered with a packet sequence number. The system for controlling the egress buffer saturation comprises for each data packet flow means for comparing the number of data packets 'WPC' temporarily stored within the egress buffer to a predefined threshold value 'WPCth'. It also comprises means for storing the packet sequence number 'PSNr' of a last received in-sequence data packet, and each highest packet sequence number 'HPSNj' respectively received through the plurality of switching planes. Finally, means that are coupled to the comparing means and to the storing means allows to determine which switching plane(s) among the plurality of switching planes on which to unstop the flow of data packets by comparing the last received in-sequence packet sequence number 'PSNr' to each highest packet sequence number 'HPSNj' when the number of data packets 'WPC' exceeds the predefined threshold value 'WPCth'.

Figure 2

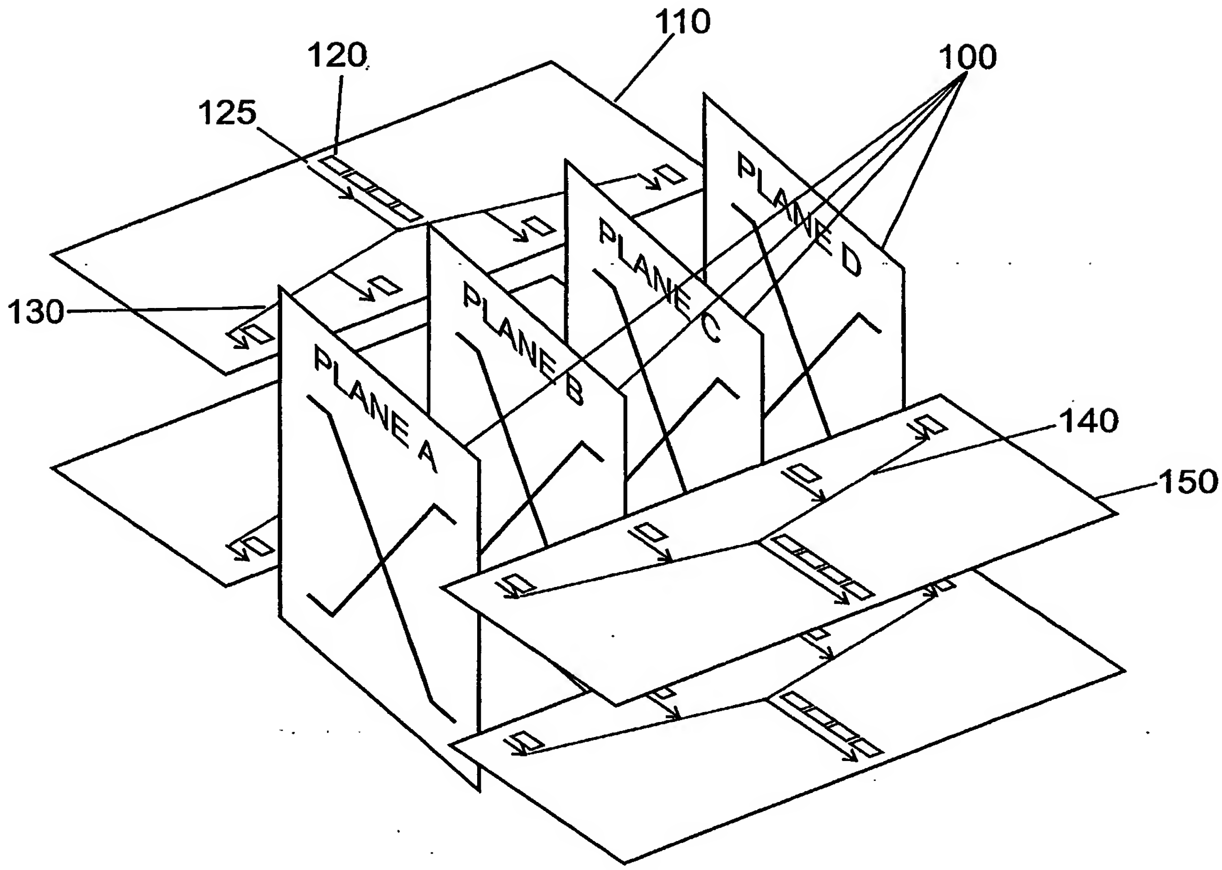


FIG. 1

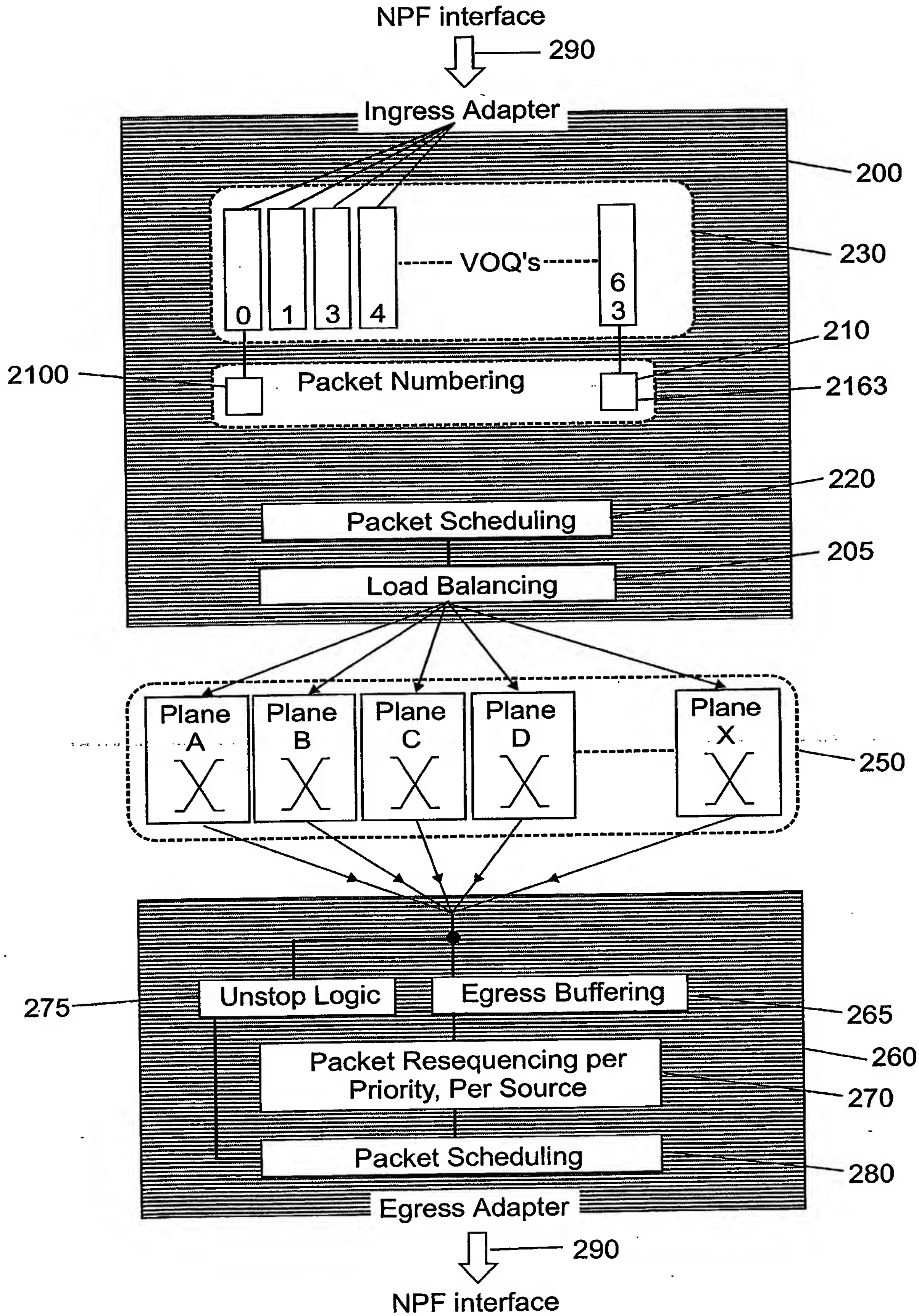

NPF interface
290
Ingress Adapter
200
230
VOQ's
0
1
3
4
6
3
210
2100
Packet Numbering
2163
220
Packet Scheduling
205
Load Balancing
Plane A
Plane B
Plane C
Plane D
Plane X
250
275
Unstop Logic
Egress Buffering
265
260
Packet Resequencing per Priority, Per Source
270
Packet Scheduling
280
Egress Adapter
290
NPF interface

FIG. 2

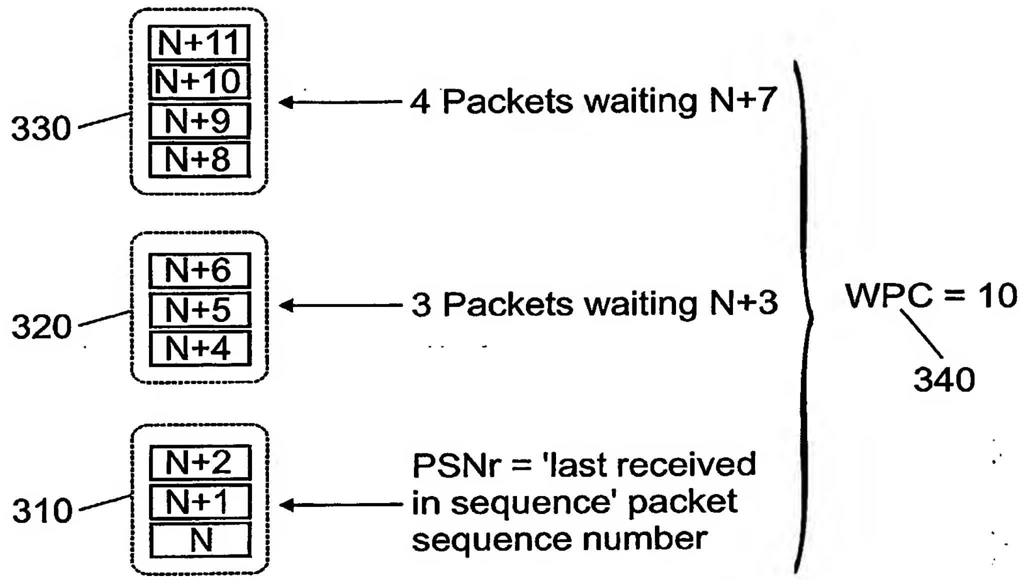


FIG. 3

410

Source, Priority	WPC	PSNr	HPSN1	HPSN2	HPSN3	HPSN4	HPSN5	HPSN6
Source, Priority	WPC	PSNr	HPSN1	HPSN2	HPSN3	HPSN4	HPSN5	HPSN6
Source, Priority	WPC	PSNr	HPSN1	HPSN2	HPSN3	HPSN4	HPSN5	HPSN6
Source, Priority	WPC	PSNr	HPSN1	HPSN2	HPSN3	HPSN4	HPSN5	HPSN6
Source, Priority	WPC	PSNr	HPSN1	HPSN2	HPSN3	HPSN4	HPSN5	HPSN6
Source, Priority	WPC	PSNr	HPSN1	HPSN2	HPSN3	HPSN4	HPSN5	HPSN6
Source, Priority	WPC	PSNr	HPSN1	HPSN2	HPSN3	HPSN4	HPSN5	HPSN6
Source, Priority	WPC	PSNr	HPSN1	HPSN2	HPSN3	HPSN4	HPSN5	HPSN6

FIG. 4

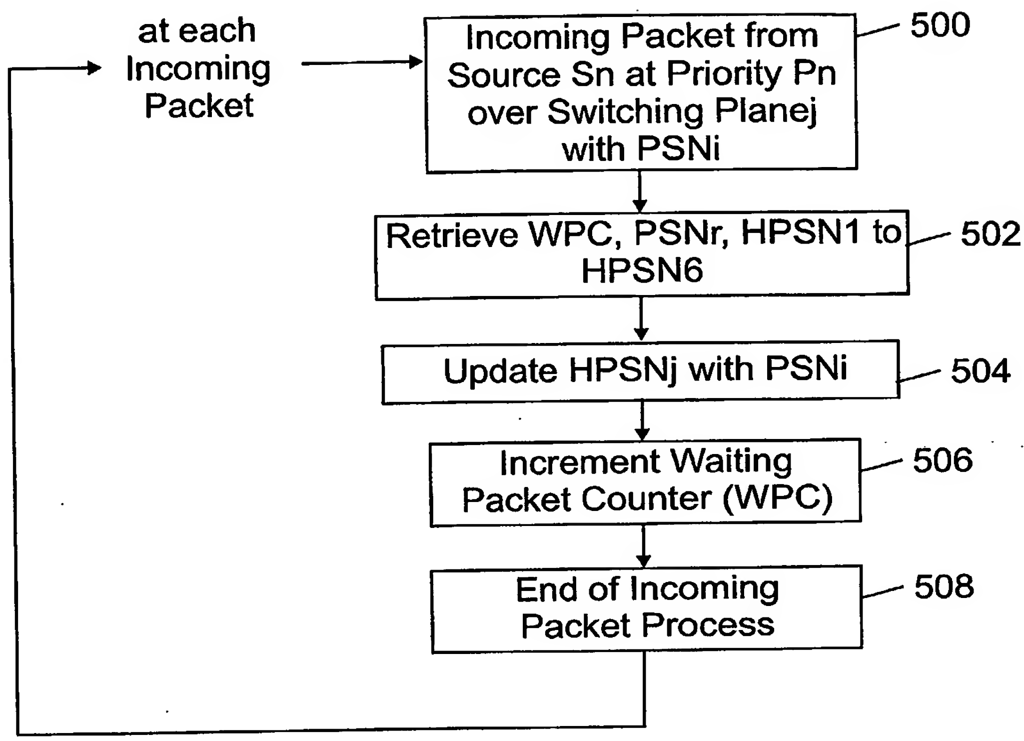

at each Incoming Packet
Incoming Packet from Source Sn at Priority Pn over Switching Planej with PSNi
500
Retrieve WPC, PSNr, HPSN1 to HPSN6
502
Update HPSNj with PSNi
504
Increment Waiting Packet Counter (WPC)
506
End of Incoming Packet Process
508

FIG. 5

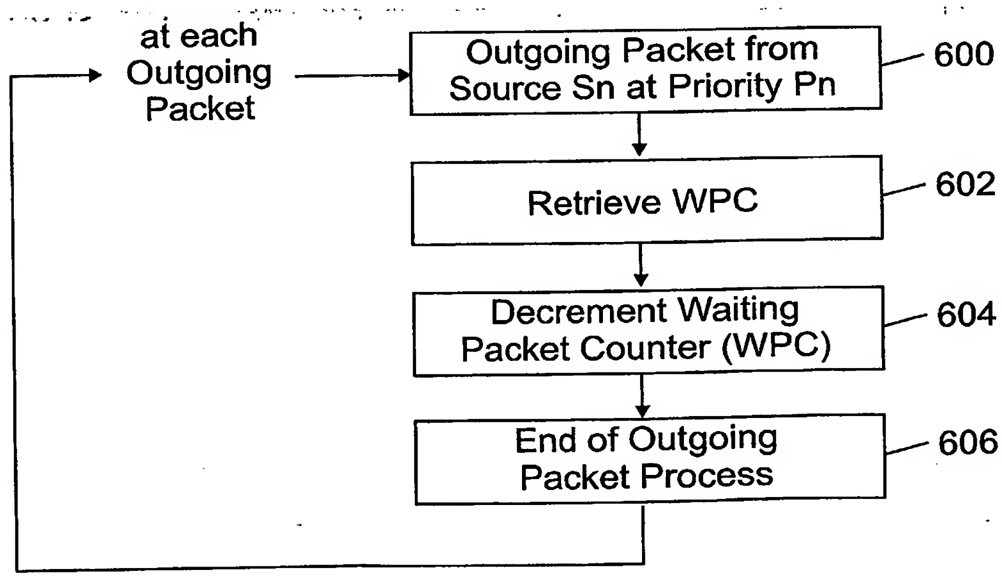
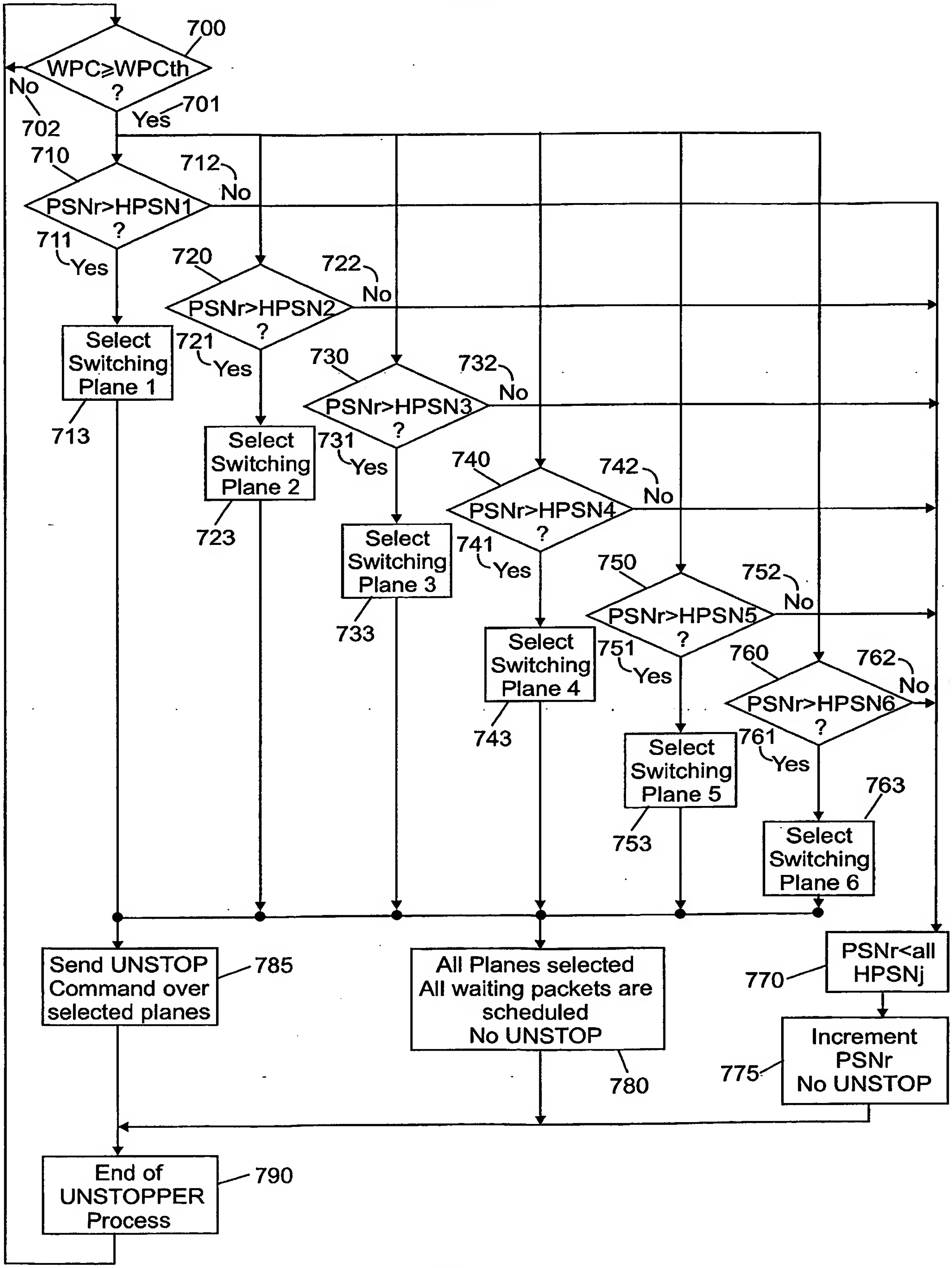

at each Outgoing Packet
Outgoing Packet from Source Sn at Priority Pn
600
Retrieve WPC
602
Decrement Waiting Packet Counter (WPC)
604
End of Outgoing Packet Process
606

FIG. 6



WPC≥WPCth ?
700
No
702
Yes
701
710
PSNr>HPSN1 ?
711
Yes
712
No
Select Switching Plane 1
713
720
PSNr>HPSN2 ?
721
Yes
722
No
Select Switching Plane 2
723
730
PSNr>HPSN3 ?
731
Yes
732
No
Select Switching Plane 3
733
740
PSNr>HPSN4 ?
741
Yes
742
No
Select Switching Plane 4
743
750
PSNr>HPSN5 ?
751
Yes
752
No
Select Switching Plane 5
753
760
PSNr>HPSN6 ?
761
Yes
762
No
Select Switching Plane 6
763
Send UNSTOP Command over selected planes
785
All Planes selected All waiting packets are scheduled No UNSTOP
780
PSNr<all HPSNj
770
Increment PSNr No UNSTOP
775
End of UNSTOPPER Process
790

FIG. 7

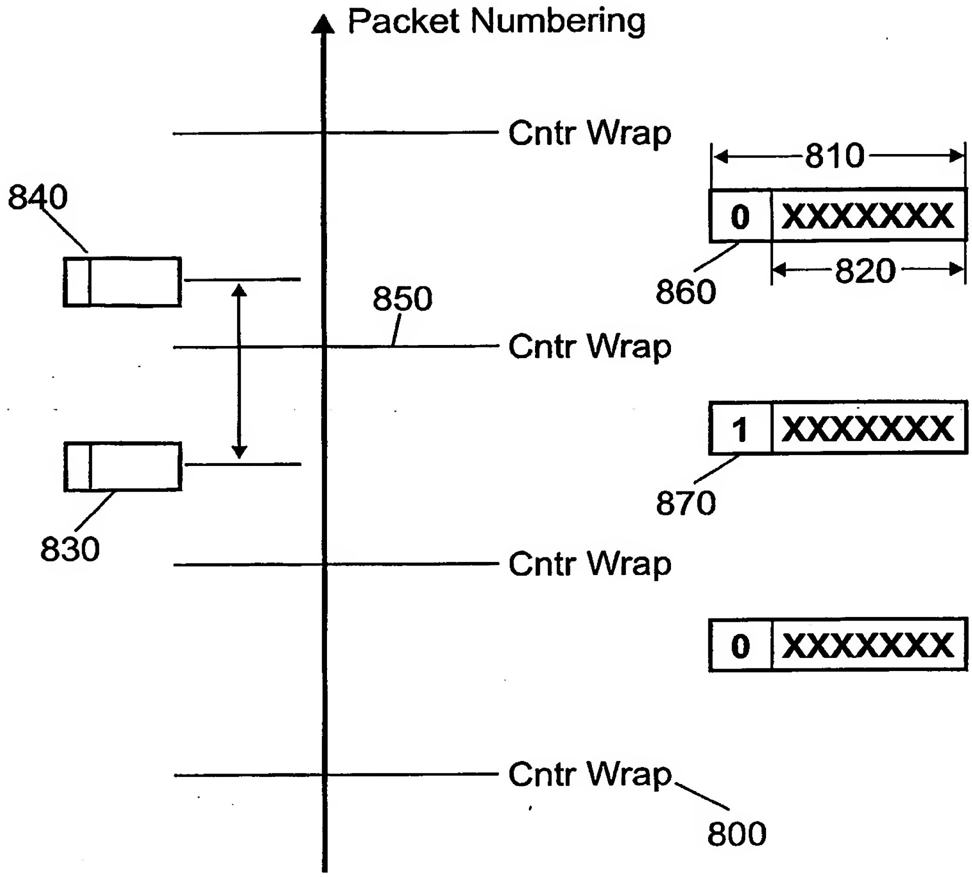

Packet Numbering
Cntr Wrap
810
0 XXXXXXX
820
860
840
850
Cntr Wrap
1 XXXXXXX
870
830
Cntr Wrap
0 XXXXXXX
Cntr Wrap
800

FIG. 8